Journal of Engineering Science and Technology
Vol. 20, No. 6 (2025) 2195 - 2218

ENERGY-EFFICIENT MAC PROTOCOL AND SCALABLE COMMUNICATION SYSTEMS IN WSN-IOT

EKHLAS K. HAMZAA[1], EMAN K. IBRAHEEM[2], AMJAD J. HUMAIDI[1], ARIF A. ALQASSAR[1]

[1]University of Technology, Iraq, Baghdad, Iraq
[2]Alkarkh University, Iraq, Baghdad, Iraq
*Corresponding Author: 60023@uotechnology.edu.iq

Abstract

The Internet of Things (IoT) is an innovative technology. It has revolutionized the world of technology today. IoT connects numerous sensors and devices to the Internet to perform various functions every day. These sensors have a finite battery life and must operate sustainably and efficiently in IoT networks. Due to these connections, the IoT consumes a significant amount of energy. Research into energy-efficient and energy-saving techniques is thus essential. This paper conducts a comparative study between the Dijkstra algorithm and Artificial Ant Colony (ACO) for energy efficiency in Wireless Sensor Networks (WSNs). This novelty lies in its systematic evaluation of deterministic and stochastic routing algorithms within identical WSN topologies. Simulation results show that Dijkstra's algorithm reduces energy consumption by up to 30% compared to ACO and achieves real-time path calculations (0.001 s) even in large networks of 100 nodes. Conversely, ACO shows greater variability in performance but can still attain optimal solutions with properly adjusted parameters. These findings provide crucial insights for choosing routing techniques in energy-constrained IoT systems. Simulations are run using settings consisting of 20, 50, and 100 sensor nodes. The levels of energy are initialized randomly from 0 to the initial capacity of the battery of (3x2.85 Ampere-hour), the effective distance range is 100 meters, and the packet size is 128 bytes.

Keywords: Artificial ant colony, Dijkstra algorithm, Energy consumption, Internet of Things, Wireless sensor networks.

1. Introduction

In recent years, IoT technologies have advanced to support a wide range of uses, with numerous applications being developed through the IoT. these technologies can overcome all challenges in fields like processing capacity, security, and data mobility, along with growth related to other technologies, to maintain their predicted path. So, IoT is described as a network of regular objects like smartphones, Internet TVs, actuators, and sensors intelligently connected to make it possible for objects to communicate with one another and with people in new ways [1-5].

WSNs have a major role in IoT technologies. All Technologies are moving towards designing wireless sensor nodes distinguished by fast CPUs and low-power radio connectivity to contribute to wireless sensor networks and IoT applications. A wide range of services and activities in daily life may be carried out using the WSNs. For instance, monitoring is a prevalent WSN use. WSN is deployed over a region to detect a specific phenomenon in the monitoring area. A military application of sensors to identify hostile intrusions could be a practical application of such a network. Suppose sensors detect an event (such as a change in blood pressure or heat). In that case, the event is promptly reported to the base station, which determines the proper course of action (such as sending a message to a satellite or the internet).

Air pollution monitoring, in which WSNs are utilized in multiple cities to monitor concentrations of hazardous gases for citizens, may be a similar application area. Additionally, a WSN may be employed in order to detect forest fires and regulate them once they have commenced. The nodes will be provided with sensors to regulate humidity, temperature, and gases produced by fires in vegetation or trees [6-8].

Furthermore, the healthcare sector is a significant application area. In this area, the WSNs have the potential to provide substantial cost savings and facilitate new functionalities that will aid elderly individuals who reside at home or individuals who have chronic diseases in their day-to-day activities. The wiring cost frequently restricts the number of sensors that can be installed in wired systems. Wireless sensors can now access locations that were formerly inaccessible, rotating devices, hazardous or restricted areas, and mobile assets [9].

WSN use in agriculture can be advantageous to the industry, as it relieves farmers from the responsibility of maintaining wiring in challenging environments. Pressure transmitters can be employed to monitor the levels of water tanks in gravity-feed water systems. Wireless I/O devices may be used in order to control pumps, and water consumption can be measured and transmitted wirelessly to a central control centre for invoicing. The water industry may benefit from the monitoring of power or data transmission through the use of industrial wireless I/O devices and sensors that are powered by battery packs or solar panels [10].

WSN comprises numerous minor devices called “sensor nodes”. Those nodes can monitor, sense, and collect data from the surroundings. The gathered data will be processed and directed to the target node (base station) using routing protocols [11]. A certain amount of energy will be consumed during these processes. These nodes have batteries and energy. If the sensor battery dies and the power supply is not replaced quickly, it becomes disabled. This will hinder the transfer of the data that

has been gathered. Consequently, one major problem in real-world remote sensing applications has been to extend the lifespan of the entire WSN [12, 13].

The selection of routing algorithms is crucial for efficiently delivering packets to their destination. Furthermore, in such networks, the implemented routing technique must guarantee minimal energy consumption, hence maximizing the network's lifespan [14]. To optimize overall network energy usage, a number of researchers and academics have conducted several studies, as will be shown in Section 2.

The key contributions of the present study can be summarized as follows:

- Comparative analysis of Dijkstra and ACO algorithms for routing in IoT-based WSNs, emphasizing energy economy and computing complexity.
- Comprehensive simulation analysis across various network sizes (20, 40, and 100 nodes), highlighting scalability effects on algorithm performance.
- Demonstration of Dijkstra's improved efficiency in reducing energy usage (up to 30% lower) and computational time (up to 20× faster) relative to ACO.
- Insights into selecting energy-efficient routing strategies for sustainable IoT-based WSNs.

The organization of the paper is as follows: Section 2 introduces recent related works on the IoT, WSNs, and energy efficiency. Section 3 describes the problem modelling and case study. Section 4 presents the suggested methods for improving energy efficiency in WSNs with the use of the Dijkstra algorithm and ACO, which will be discussed in this paper. Section 5 details simulation settings and outcomes. Section 6 shows the conclusion.

2. Related Works

Kaur and Mahajan [15] proposed a hybrid ACOPSO-based clustering protocol to enhance energy efficiency in wireless sensor networks (WSNs). This protocol segments a WSN into multiple clusters, selecting cluster heads for each one. After that, the sensing data is collected directly from each cluster head by tree-based data aggregation using short-distance connections. The suggested ACOPSO protocol selects the shortest path from the sink node to every CH. To test and evaluate the proposed method's performance with the current technique (GSTEB) using the following metrics: network lifetime, throughput, stability period, and residual energy. A MATLAB simulation involves 100 sensor nodes randomly allocated in a 100x100 area, utilizing a single base station. The results indicate that the proposed hybrid protocol significantly enhances the lifespan of the WSN.

Abderrahim et al. [16] proposed an energy-conserving algorithm using the Dijkstra algorithm, which is applied to WSN after it has been segmented into several clusters, so that each node is assigned to the nearby cluster head (CH) based on the distance to the CH. According to their distance from the source node, CH is in charge of informing nodes in its cluster to be sleeping or active. The proposed path selection algorithm selects a list of appropriate paths to transmit data from the source node to the CH with the lowest energy cost.

Zhang et.al. [17] suggested a model consisting of sensor nodes, one macro base station, and several micro base stations. The micro base station transmits the data that it collects from sensor nodes to the macro base station. The total WSN's energy consumption is reduced using the suggested Restart Artificial Bee Colony (RABC)

algorithm. The suggested model consists of 3 stages for optimization problems. These stages are used to select the optimal uplink route, find the shortest travel path, and for mutual interference. The simulation outcomes found optimal and near-optimal solutions by the RABC method.

Jain and Agrawal [18] enhanced the LEACH protocol to solve the sensor's battery limitation and the WSN lifetime problem. The LEACH protocol is a categorized protocol that utilizes the algorithm of Cluster Head (CH) selection and cache node selection for data transmission from the source node to the destination node. Based on comparing energy with a threshold, the CHs are chosen. Also, based on the minimum distance, the transmitted data from CH to the cache node is completed. The replacement of cache node memory is organized based on priority. They evaluate the performance of the proposed approach based on several standard parameters such as throughput, average residual energy, and live or dead nodes in the network.

Bhola et al. [19] enhance energy efficiency and longevity of WSNs. This study proposes the LEACH routing protocol in conjunction with the genetic algorithm (GA). LEACH is a hierarchical protocol that identifies CH within a wireless sensor network; the CH aggregates and compresses data before transmitting it to the base station node. The ideal path is determined by the Genetic Algorithm, which utilizes its fitness function. Results from the MATLAB simulator indicate an energy consumption differential of up to 17.39% between the proposed methodology and existing contemporary methods.

Prajapat et al. [20] proposed for cognitive radio sensor networks (CRSN) a neighbour finding algorithm and 2 greedy k-hop clustering schemes (k-SACB-EC, and k-SACB-WEC). The primary objective of this study is to enhance network lifetime and achieve bi-channel connectivity. Several parameters are considered, such as nodes' primary users (PUs) appearance probability of channels, residual energy, channel quality, spectrum awareness, and robustness on PUs' arrival. Simulation is used to evaluate the suggested method concerning the network lifespan, number of clusters, network stability, and re-clustering frequency. The k-SACB-WEC generates several clusters fewer than NSAC, k-SACB-EC, PSEP, CogLEACH, and SAC-WCM by 40%. Moreover, the k-SACB-WEC attains a minimum of around 100% greater number of rounds before the first node dies than the related methods concerning network stability.

Abdulzahra et al. [21] proposed a Bacterial foraging optimization routing protocol (BFORP) as a routing protocol for energy-saving to explore the problem of the WSNs' lifespan. By reprocessing the data that visits the source node frequently to the sink, it can lead to reducing the routing of extreme messages that may cause waste of energy. In the proposed approach, the lowest traffic load, the shortest path to the sink, and the maximum residual energy are considered to select nodes for the sending routes. The simulation outcomes proved the efficiency of the proposed protocol in reducing the consumption of energy and end-to-end delay.

Hamza et al. [22] improved the Grey Wolf method with Particle Swarm Optimization and Tabu Search Techniques (GW-IPSO-TS) to enhance the Cluster Heads (CHs) selection according to the CH probabilistic reasoning of sensor nodes, also enhancing the routing path of each cluster head to the base station. The suggested model delivered the optimal routing paths and enhanced the overall WSNs' lifetime, the packet loss rate, and end-to-end delay.

Gunigari and Chitra [23] suggested a reliable and energy-saving hybrid method, based on ACO, E-RARP routing protocol, and game theory clustering algorithm (GEC). Reliable communications, reliability, and high-quality communication channels are provided by the proposed protocol to enhance energy consumption. every sensor node is considered a team player in WSNs using a game theory clustering algorithm (GEC), so it can select for itself a beneficial method, defined by the duration of inactive playback time in the active phase, and afterward choose whether to rest or not. The suggested E-RARP-GEC enhanced WSNS' lifetime; it also consumes a minimal amount of energy when compared to other proposed methods.

Several studies have employed hybrid approaches to enhance WSNs' efficiency and longevity. However, these methods can be computationally demanding, which may impede resource-constrained sensor networks. The GW-IPSO-TS method combines Grey Wolf Optimization, PSO, and Tabu Search, thereby increasing the computations for cluster head selection and routing path optimization [22]. The game-theory-based clustering strategy requires each sensor node to make decisions independently, leading to increased decision-making overhead [23].

The Restart Artificial Bee Colony (RABC) algorithm implements a multi-stage optimization procedure that is efficient but demands more computing power and memory [17]. While these complex hybrid methods enhance performance, they also impose significant computational and energy burdens, rendering them unsuitable for large-scale WSN deployments [24]. Hence, there is a need for a more balanced approach that maintains optimization advantages while reducing complexity. Table 1 presents a summary of the related works discussing potential limitations of the proposed methods.

Table1. Summary of related works.

No.	Research	Year	Algorithms	Limitations
1	Kaur and Mahajan [15]	2018	• ACO • PSO	• The protocol can become more complex when two optimization algorithms (ACO and PSO) are combined.
2	Abderrahim et al. [16]	2019	• Dijkstra	• Dijkstra's algorithm for relay selection can be inefficient for larger networks. • Maintaining clusters can lead to increased communication overhead, especially in dynamic environments.
3	Zhang et al. [17]	2019	• RABC	• Implementing the RABC method in real-world scenarios can be complex.
4	Jain and Agrawal [18]	2020	• Improved LEACH	• Modified LEACH protocol may face challenges managing cache nodes due to increased complexity, resource demands, and scalability.
5	Bhola et al. [19]	2020	• GA-based LEACH	• Inherits LEACH's weaknesses like random cluster sizes and single-hop communication,

			optimization in WSN	reducing effectiveness in large or dynamic networks. • Optimization introduces computational cost; GA parameters were not thoroughly analyzed.
6	Prajapat et al. [20]	2021	• k-SACB-EC • k-SACB-WEC	• The comprehensive consideration of multiple parameters poses potential complexity and resource demands that may challenge IoT deployment feasibility and scalability.
7	Abdulzahra et al. [21]	2023	• BFORP	• BFORP relies on idealized conditions for node selection, which limits its effectiveness. It may struggle in dynamic environments where network conditions change rapidly.
8	Hamza et al. [22]	2023	• Particle Swarm • Grey Wolf • Tabu Search	• The GW-IPSO-TS method's main limitation is its complexity and computational overhead due to multiple optimization techniques.
9	Gunigari and Chitra [23]	2023	• ACO • GEC	• The integration of E-RARP and GEC techniques in the proposed system may lead to increased complexity and overhead.

3. Proposed Methodology and Case Study

Previous work has often encountered issues such as scalability and overhead issues in dynamic or large-scale environments, limitations in real-world applicability because of idealized assumptions or inadequate parameter tuning, and increased complexity because of the integration of multiple optimization algorithms. Dijkstra's algorithm and ACO were selected for this study because of their divergent methodologies and extensive utilization in pathfinding and optimization challenges. Dijkstra's algorithm is a traditional deterministic technique recognized for its precision and efficacy in identifying the shortest path in graphs with non-negative edge weights.

Conversely, ACO is a bio-inspired metaheuristic that provides flexibility and adaptation, rendering it appropriate for dynamic and intricate situations. The scenario studied in this paper is about WSNs in the greenhouse. WSNs in greenhouses are used to monitor crucial environmental factors for plant growth. Such as for a smart irrigation system, a WSN can be utilized to monitor the field and control watering status [20, 21, 25]. The WSNs comprise 20, 40, and 100 sensor nodes, along with a central base station.

3.1. Network setup

Node Positions: Sensor nodes are strategically placed to effectively cover the greenhouse area, one sensor per greenhouse. Different topologies are used for 20, 40, and 100 nodes.

Base Station: Strategically situated to receive and process data sent by sensor nodes. Sensors are distributed across greenhouses to gather data and transmit it to a BS for monitoring via multi-hop communication, as shown in Fig.1 for 20 and 40 topologies. While Fig. 2 illustrates the topology of WSNs that connect 100 greenhouses.

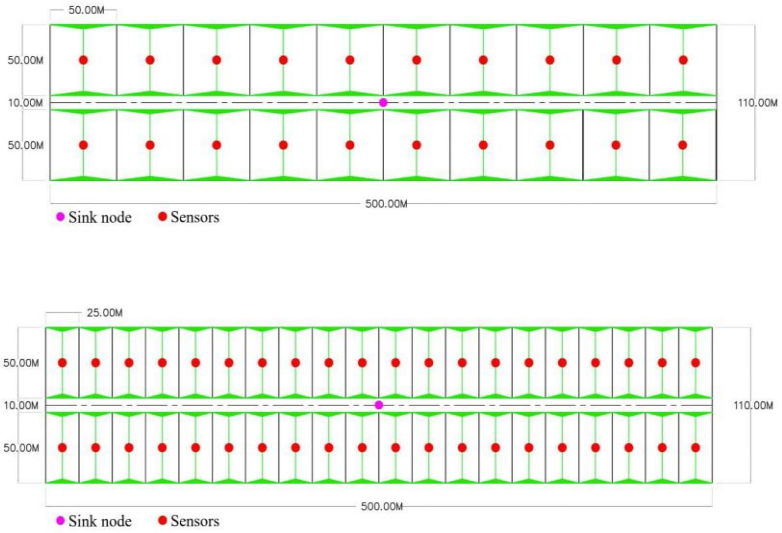


Fig. 1. WSN topology that connects the 20 and 40 greenhouses.

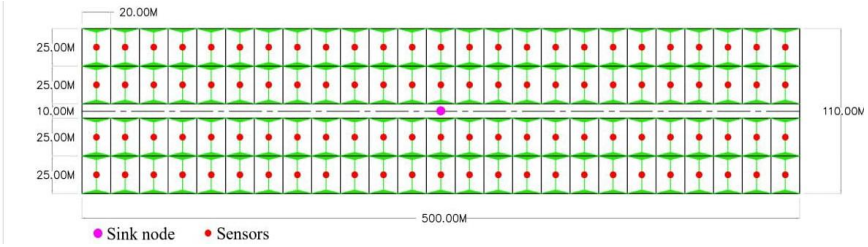


Fig. 2. WSN topology that connects the 100 greenhouses.

3.2. Problem statement

The key limitation of WSNs is the battery capacity of sensor nodes. So, the main challenge is to decrease energy consumed by WSNs while ensuring efficient data transmission from sensor nodes to the base station.

3.3. Objectives

This paper evaluates and compares the performance of two algorithms (ACO and Dijkstra's algorithm) for energy efficiency for routing data in greenhouses' WSNs. It suggests different network topologies (20, 40, and 100 nodes) and discusses their energy consumption and routing efficiency.

Additionally, the primary objective of this work is to determine the most energy-efficient method between the two proposed methods, which enables reliable data transmission with minimal energy consumption. The energy consumption of a

single node throughout the transmission can be estimated with the equations (1-3) [26, 27].

$$E_{tx} = (E_{elec} + E_{amp} \times d^2) \times L \quad (1)$$

E_{elec} denotes the energy (expressed in Joules per bit) used by the transmitter or reception electronics. E_{amp} (expressed in Joules/bit/m²) represents energy that is required to transport a bit wirelessly, L denotes the quantity of bits sent, and d signifies the distance between nodes. The energy used by a node during reception may be determined utilizing Eq. (2).

$$E_{rx} = E_{elec} \times L \quad (2)$$

So, the energy that is used by a single node is:

$$E_{node} = E_{tx} + E_{rx} \quad (3)$$

Equation (4) provides the cost function, which represents the selected path's total energy consumption.

$$E_{total} = \sum_{i=1}^{n} E_{node}(i) \quad (4)$$

4. Proposed Energy Efficient Methodology

The presented method for energy efficiency in WSNs is based on 2 algorithms, which are, Dijkstra algorithm and Ant Colony optimization algorithm, to select the minimum energy cost path between the source node and BS. The proposed methodology is represented in the block diagram in Fig 3.

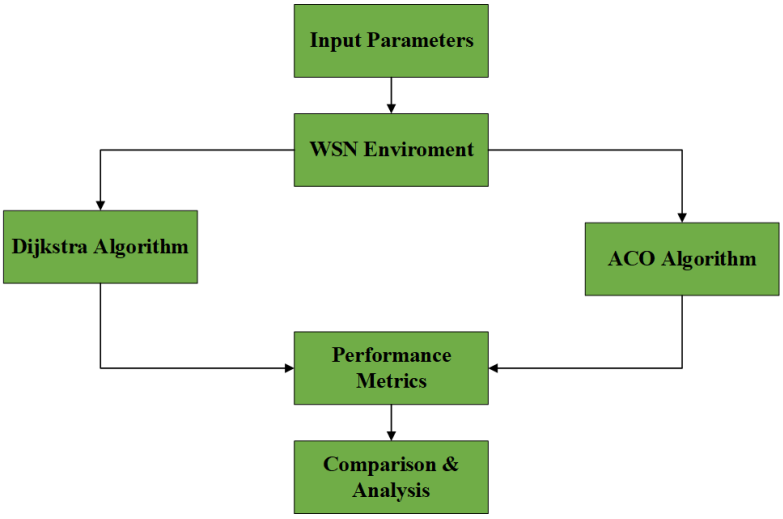


Fig. 3. General block diagram for the proposed methodology.

4.1. The Dijkstra algorithm

The Dijkstra algorithm is a widespread algorithm to resolve shortest-path problems using graphs with non-negative edge weights. It is used to obtain shortest path between a pair of vertices on a graph with minimum distance cost. It has been created by computer scientist Edger W. Dijkstra in 1956 [28].

The algorithm does its computation using a set of visited and a set of unvisited vertices of the graph. It begins at source vertex and iteratively chooses from the unvisited set a vertex with minimum distance to the source. After that, the neighbours of this vertex are visited, and their tentative distances are updated if a shorter path is discovered. This procedure is continued until either all reachable vertices have been visited, or the destination vertex has been reached [29, 30].

The Dijkstra method iteratively tracks a distance value for every node in the weighted network. Except for the source node (s), all other nodes have these edge distances set mostly to infinity. This convention indicates that certain vertices haven't been visited yet, but it doesn't mean the distance is limitless. There is a "tree" with root (s) at each iteration. In the first iteration, only the source node (s) are in the tree, with no distance to it. The next node (v), which is the closest to (s), will then be added to the tree [31-33].

To determine whether the path from (s) to (v) and (u, v) edge leads to a shorter distance to (u) than what has previously been stored, v is added to the tree and its neighbours, including u, are examined. If the response is positive, the distance of (u) is updated to a lesser value. For every vertex v, a parent vertex determines the final edge required to reach v from s.

In a weighted graph G = (V, E), edges can be mapped to real-valued weights using weight function w: $E \rightarrow R$. W is expressed as $w\ (u, v)$ for $w(e)$ if $e = (u, v)$. The shortest path between a pair of vertices is known as link distance. $\delta\ (u, v)$ indicates the distance from u to v, or the shortest way from u to v if there is a path and is ∞ otherwise. The shortest path from node a to node e in the graph depicted in Fig. 4 is the path (a, b, c, e), and the shortest path is also the sub-path (a, b, c). Weighted graphs can be used to illustrate wireless sensor networks to determine the optimal route between a source node and a destination node [29, 33].

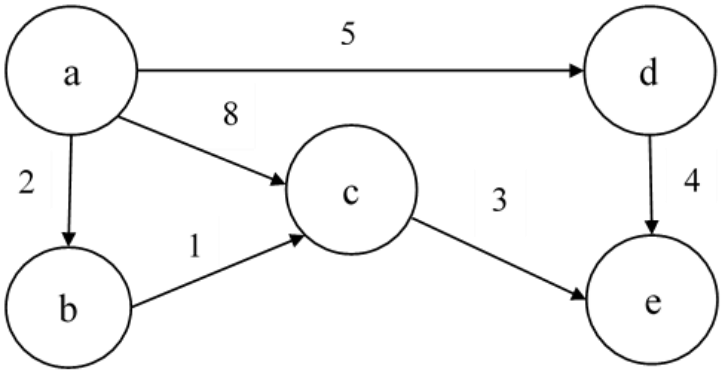


Fig. 4 A weighted graph with 5 nodes.

In this work, the Dijkstra algorithm considers the battery level of each node while selecting the best path with minimum energy cost. It is vital to keep away disabled or low battery-level nodes. Dijkstra's method was used for this investigation because it is deterministic and low-complexity, making it optimal for wireless sensor networks with limited resources. Dijkstra calculates accurate shortest paths without the need for random exploration or iterative convergence, in contrast to metaheuristic algorithms like ACO.

4.2. Ant colony optimization algorithm (ACO)

ACO is a nature-inspired algorithm of optimization developed in the mid-1990s by Marco Dorigo. It uses artificial ants to mark paths with pheromones, intensifying trails on shorter paths to guide exploration toward more efficient routes. The main ACO steps are illustrated below [23, 25, 34-36]:

Step1: Initialization:

a. Pheromone trails initialization: to initialize the pheromone levels, assign values to the edges of the graph.

b. Ant positions initialization: Place artificial ants on source nodes.

Step2: Ant Movement:

a. Ant Selection: Ants probabilistically choose their next movement based on pheromones and heuristics.

The probability of k^{th} ant at i^{th} node choosing j^{th} node as the next node is expressed in Eq. (5):

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{i \in \text{allowed nodes}} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \quad (5)$$

where: τ_{ij}: Pheromone level on edge (i →j), η_{ij}: Heuristic information, and parameters α and β are utilized to regulate the impact of pheromone and heuristic data.

Step3: Move Ants

Artificial ants select their next node based on probabilities.

Pheromone Update:

a. Pheromone Evaporation: for simulating evaporation, decrease the levels of pheromone on all edges as shown in Eq. (6):

$$\tau_{ij} = (1 - \rho).\tau_{ij} \quad (6)$$

where ρ represents the Pheromone evaporation rate and ($0 < \rho < 1$).

b. Pheromone Deposition: Ants leave a trail of pheromones along the edges they travel, which is based on the solution quality.

Ant *k* traverses' edge (i→j), and the pheromone level on this edge is updated using Eq. (7):

$$\Delta\tau_{ij}^{k} = \frac{1}{total\ cost\ chosen\ by\ ant\ k} \quad (7)$$

The total cost in this work is the total energy consumption chosen by ant k.

c. Global Update: to update the pheromone levels on edges, taking into account all the ant solutions as expressed in Eq. (8):

$$\tau_{ij} = (1 - \rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (8)$$

where m is the number of ants.

Step4: Termination

Iterate Ants movement, pheromone update, and solution construction till a termination condition is met (e.g., reach the max iteration number). ACO adjusts pheromone levels on edges to bias ant movements toward high concentrations, promoting the exploration of efficient routes while exploiting prior information.

5. Simulations and Results Analysis

In this section, the simulation setup and the main results conducted by the proposed methodology are presented.

5.1. Simulation setup

The simulations aimed to assess the energy efficiency of WSNs using Dijkstra's algorithm and Ant Colony Optimization (ACO). The experiments were conducted on WSNs of varying sizes, including 20, 40, and 100 sensor nodes, to analysed performance across different network scales comprehensively. The metrics evaluated were energy consumption, path optimality, and computational time.

Tables 2-4 display the positions of nodes in the WSN topologies of sizes 20, 40, and 100, respectively. The levels of energy are initialized randomly within the range from 0 to the initial battery capacity of 3x2.85 Ampere-hour. The data rate is 250,000 bps, the distance range is 100m, and the transmit power is 1.25 milliwatt in normal mode. Lastly, the packet size is 128 bytes.

Table 2. Coordinates of 20 sensor nodes in the 20-topology WSN.

Sensor	X	Y	Sensor	X	Y
1	25	25	12	75	85
2	75	25	13	125	85
3	125	25	14	175	85
4	175	25	15	225	85
5	225	25	16	275	85
6	275	25	17	325	85
7	325	25	18	375	85
8	375	25	19	425	85
9	425	25	20	475	85
10	475	25	Sink Node	250	55
11	25	85			

Table 3. Coordinates of 40 sensor nodes in the 20-topology WSN.

Sen.	X	Y	Sen.	X	Y	Sen.	X	Y	Sen.	X	Y
0	12.5	25	11	287.5	25	23	87.5	85	34	362.5	85
1	37.5	25	12	312.5	25	24	112.5	85	35	387.5	85
2	62.5	25	13	337.5	25	25	137.5	85	36	412.5	85
3	87.5	25	14	362.5	25	26	162.5	85	37	437.5	85
4	112.5	25	15	387.5	25	27	187.5	85	38	462.5	85
5	137.5	25	17	437.5	25	28	212.5	85	39	487.5	85
6	162.5	25	18	462.5	25	29	237.5	85	Sink Node	250	55
7	187.5	25	19	487.5	25	30	262.5	85			
8	212.5	25	20	12.5	85	31	287.5	85			
9	237.5	25	21	37.5	85	32	312.5	85			
10	262.5	25	22	62.5	85	33	337.5	85			

Table 4. Coordinates of 100 sensor nodes in the 20-topology WSN.

Sensor	X	Y	Sensor	X	Y	Sensor	X	Y
0	10	12.5	35	210	37.5	70	410	72.5
1	30	12.5	36	230	37.5	71	430	72.5
2	50	12.5	37	250	37.5	72	450	72.5
3	70	12.5	38	270	37.5	73	470	72.5
4	90	12.5	39	290	37.5	74	490	72.5
5	110	12.5	40	310	37.5	75	10	97.5
6	130	12.5	41	330	37.5	76	30	97.5
7	150	12.5	42	350	37.5	77	50	97.5
8	170	12.5	43	370	37.5	78	70	97.5
9	190	12.5	44	390	37.5	79	90	97.5
10	210	12.5	45	410	37.5	80	110	97.5
11	230	12.5	46	430	37.5	81	130	97.5
12	250	12.5	47	450	37.5	82	150	97.5
13	270	12.5	48	470	37.5	83	170	97.5
14	290	12.5	49	490	37.5	84	190	97.5
15	310	12.5	50	10	72.5	85	210	97.5
16	330	12.5	51	30	72.5	86	230	97.5
17	350	12.5	52	50	72.5	87	250	97.5
18	370	12.5	53	70	72.5	88	270	97.5
19	390	12.5	54	90	72.5	89	290	97.5
20	410	12.5	55	110	72.5	90	310	97.5
21	430	12.5	56	130	72.5	91	330	97.5
22	450	12.5	57	150	72.5	92	350	97.5
23	470	12.5	58	170	72.5	93	370	97.5
24	490	12.5	59	190	72.5	94	390	97.5
25	10	37.5	60	210	72.5	95	410	97.5
26	30	37.5	61	230	72.5	96	430	97.5
27	50	37.5	62	250	72.5	97	450	97.5
28	70	37.5	63	270	72.5	98	470	97.5
29	90	37.5	64	290	72.5	99	490	97.5
30	110	37.5	65	310	72.5	Sink Node	250	55
31	130	37.5	66	330	72.5			
32	150	37.5	67	350	72.5			
33	170	37.5	68	370	72.5			
34	190	37.5	69	390	72.5			

5.2. Simulation and discussion

The energy efficiency of Dijkstra's algorithm and Ant Colony Optimization in WSNs is compared in this section. The analysis reveals significant differences in energy cost and time complexity between the two algorithms. Moreover, the paper explores various settings in ACO, such as iterations, cycles, and the number of ants, to evaluate their impact on energy efficiency within the network. The results of the three topologies (20, 40, and 100) are expressed.

Tables (5-7) show the selected results of the three topologies in different settings of the ACO algorithm for energy cost and time complexity.

Table 5 shows the selected results for the 20-node topology to select the optimal route between Sensor0 and the base station using ACO with different settings. While the output of the Dijkstra algorithm for the 20-sensor node for optimal Path from Sensor0 to Sink node is: [Sensor0, Sensor1, Sensor3, Sink node], distance cost: 233.77, time: 0.0 seconds, energy cost: 0.00226, Figs. 5 and 6 show the output of the

Dijkstra algorithm and ACO optimal paths from Sensor0 to Sink node of 20-node WSN. Fig. 7 shows different runs of the ACO algorithm under the same conditions, yielding different solutions that reflect the random behavior of the ACO.

Table 5. Selected results of the 20-node topology using different settings of the ACO concerning energy cost and time complexity

Ants	Iter	Cycles	Cost(m)	Energy	Time(s)
5	50	10	239.05	0.0021	0.0058
5	50	30	239.05	0.0021	0.0029
5	50	50	230.78	0.0023	0.0158
5	100	10	267.15	0.0021	0.0058
5	100	30	230.78	0.0023	0.0160
10	50	10	258.88	0.0026	0.0043
10	50	30	318.88	0.0031	0.0158
10	50	50	230.78	0.0023	0.0319
10	100	10	230.78	0.0023	0.0022
10	100	30	318.88	0.0031	0.0158
20	50	10	230.78	0.0023	0.0160
20	50	30	230.78	0.0023	0.0320
20	50	50	230.78	0.0019	0.0342
20	100	10	230.78	0.0019	0.0137
20	100	30	230.78	0.0023	0.0317

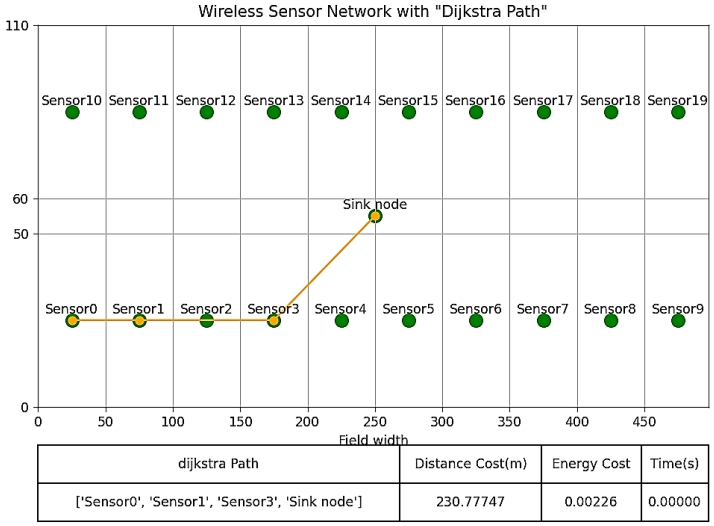
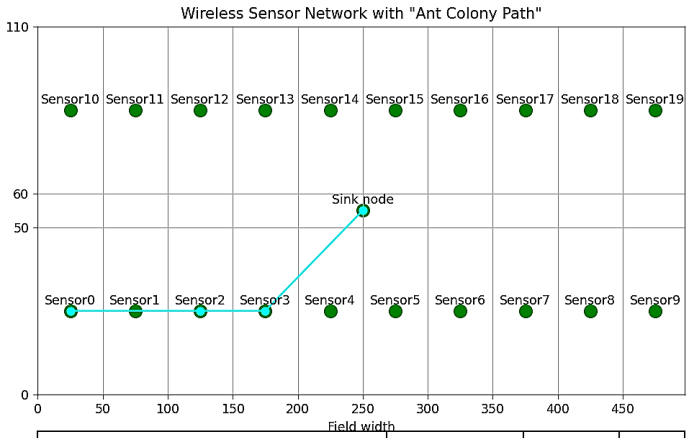


Fig. 5. Dijkstra algorithm best path from sensor0 to the sink.

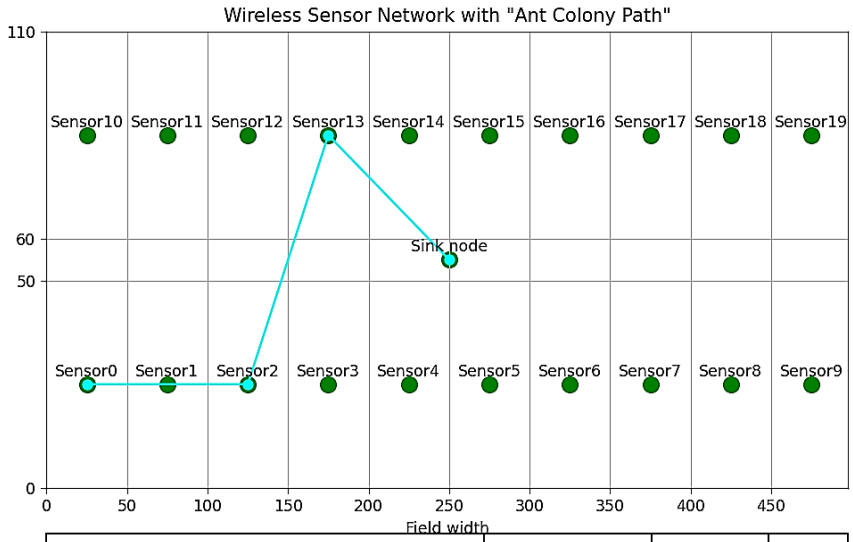
Figures 8 and 9 show the paths selected by the Dijkstra algorithm and ACO, respectively, when node 3 is disabled (turned off), illustrating the energy awareness of the two algorithms. The Dijkstra algorithm select [Sensor0, Sensor2, Sensor4, Sink node], as an optimal path with distance cost= 239.05, Time: 0.0seconds. The ACO algorithm selects the path [Sensor0, Sensor2, Sensor4, Sink node], with a distance cost of 239.05, an energy cost of 0.0251, and a time of 0.062 seconds. Table 6 shows some selected results for the 40-node topology to select the optimal route between

Sensor0 and the sink node using ACO with different settings in terms of the number, iterations and the cycle number, and their impact on selecting the best route.



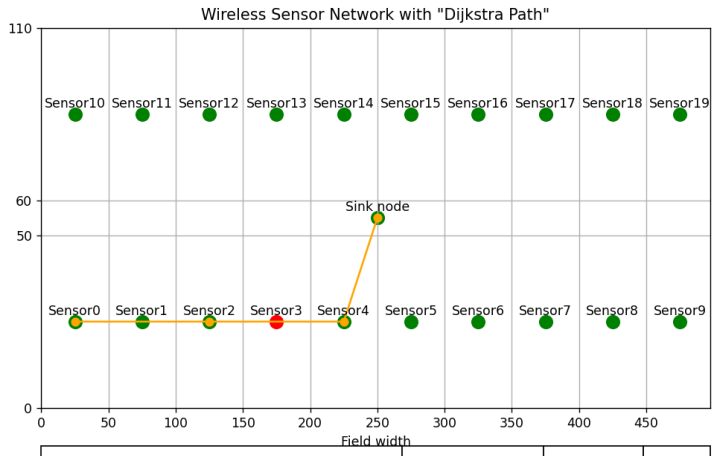
ACO Path	Distance Cost(m)	Energy Cost	Time(s)
['Sensor0', 'Sensor2', 'Sensor3', 'Sink node']	230.77747	0.00226	0.06067

Fig. 6. ACO optimal route from sensor0 to sink node.



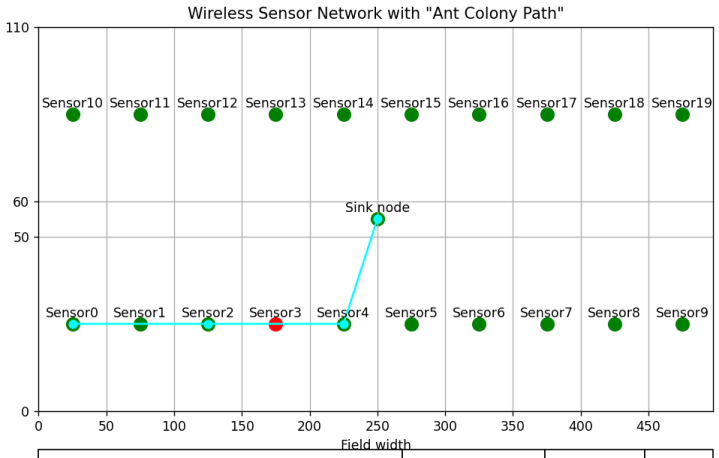
ACO Path	Distance Cost(m)	Energy Cost	Time(s)
['Sensor0', 'Sensor2', 'Sensor13', 'Sink node']	258.87997	0.00262	0.05015

Fig. 7. Near-optimal route found by ACO algorithm from sensor0 to sink node.



dijkstra Path	Distance Cost(m)	Energy Cost	Time(s)
['Sensor0', 'Sensor2', 'Sensor4', 'Sink node']	239.05125	0.00251	0.00000

Fig. 8. Dijkstra optimal route after node 3 became disabled.



ACO Path	Distance Cost(m)	Energy Cost	Time(s)
['Sensor0', 'Sensor2', 'Sensor4', 'Sink node']	239.05125	0.00251	0.06268

Fig. 9. ACO optimal route after node 3 became disabled.

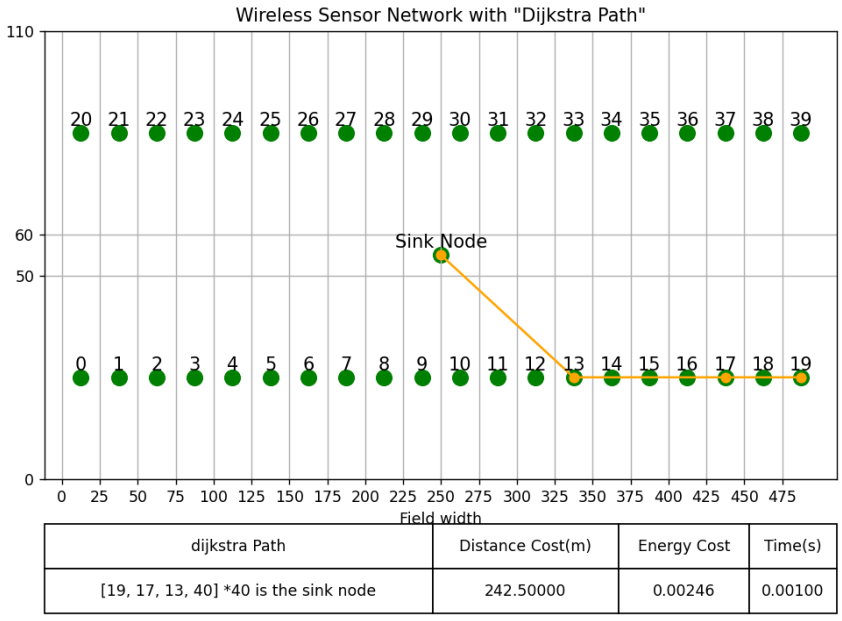
Figures 10 and 11 show the paths chosen by the Dijkstra and ACO algorithms. Dijkstra algorithm selects the path [Sensor19, Sensor17, Sensor13, Sink node], with a distance cost of 242.5, an energy cost of 0.00246, and a time of 0.001 seconds. Meanwhile, the ACO picks [19, 18, 17, 16, 15, 14, 13, Sink node], with a distance cost of 242.5, an energy cost of 0.00198, and a time of 0.439 seconds.

Figures 12 and 13 display the paths from sensor 75 to the sink node that have been chosen by the ACO and Dijkstra algorithms in a 100-node topology. The Dijkstra algorithm selects: [Sensor75, Sensor79, Sensor58, Sink node], with a distance cost of 245.7, an energy cost of 0.00237, and a time of 0.0 seconds. Meanwhile, the ACO selects a near-optimal path: [75, 76, 77, 78, 80, 82, 36, Sink node], with a distance cost of 266.5, energy cost of 0.00226, and

time of 1.15 seconds. Table 7 presents some selected results of the 100-node topology using ACO.

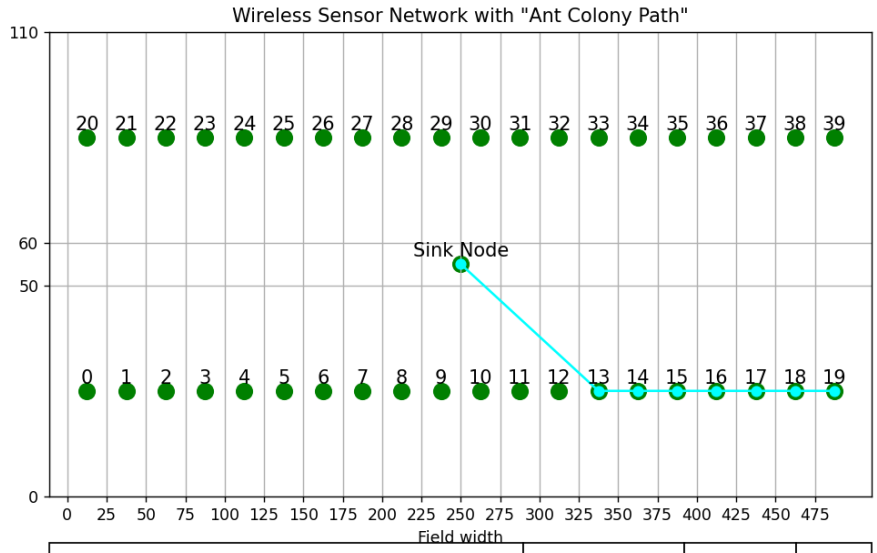
Table 6. Some selected results for The 40-node topology using the ACO algorithm.

Ants	Iter	Cycles	Cost(m)	Energy	Time(s)
5	50	10	267.15	0.0025	0.014
5	50	30	258.88	0.0026	0.000
5	50	50	239.05	0.0021	0.016
5	100	10	350.78	0.0028	0.000
5	100	30	290.78	0.0027	0.016
5	100	50	230.78	0.0023	0.016
10	50	10	230.78	0.0023	0.000
10	50	30	290.78	0.0027	0.016
10	50	50	258.88	0.0026	0.016
10	100	10	286.98	0.0026	0.016
10	100	30	290.78	0.0027	0.016
10	100	50	230.78	0.0023	0.032
20	50	10	258.88	0.0026	0.000
20	50	30	230.78	0.0023	0.032
20	50	50	299.05	0.0022	0.048
20	100	10	318.88	0.0031	0.016
20	100	30	230.78	0.0023	0.032
20	100	50	230.78	0.0023	0.063



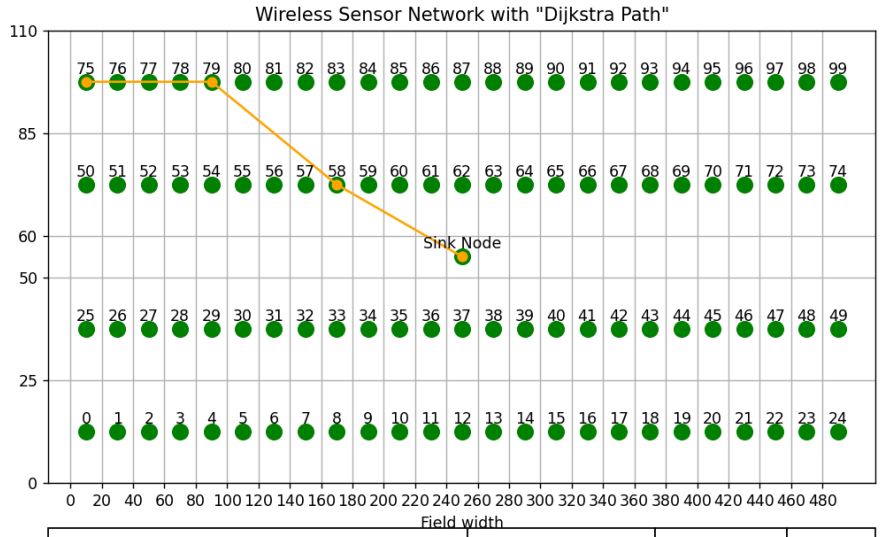
dijkstra Path	Distance Cost(m)	Energy Cost	Time(s)
[19, 17, 13, 40] *40 is the sink node	242.50000	0.00246	0.00100

Fig. 10. Dijkstra's selected route from sensor19 to the sink node in a (40-node) topology.



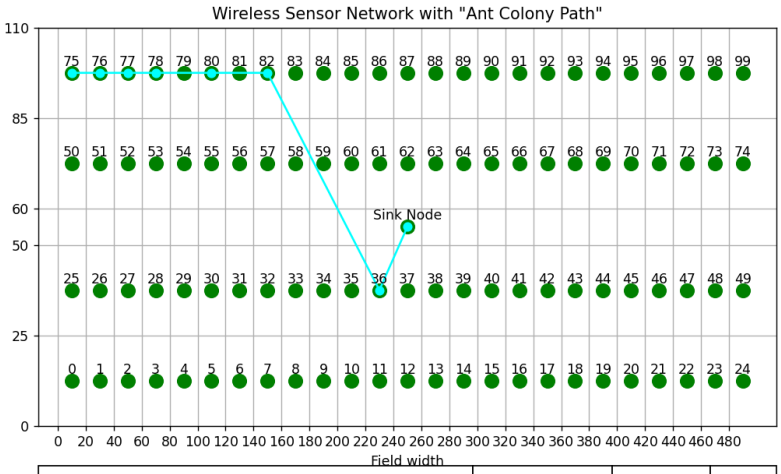
ACO Path	Distance Cost(m)	Energy Cost	Time(s)
[19, 18, 17, 16, 15, 14, 13, 40] *40 is the sink node	242.50000	0.00198	0.43900

Fig. 11. ACO selected the route from sensor19 to sink node in a (40-node) kind of topology.



dijkstra Path	Distance Cost(m)	Energy Cost	Time(s)
[75, 79, 58, 100] *100 is the sink node	245.70697	0.00237	0.00000

Fig. 12. Dijkstra's best path from sensor75 to the sink node.



ACO Path	Distance Cost(m)	Energy Cost	Time(s)
[75, 76, 77, 78, 80, 82, 36, 100] *100 is the sink node	266.57536	0.00226	1.15367

Fig. 13. ACO optimal route from sensor 75 to the sink node.

Table 7. Some selected results for 100-node topology using the ACO algorithm

Ants	Iter	Cycles	Cost(m)	Energy	Time(s)
5	50	10	239.05	0.0021	0.0036
5	50	30	258.88	0.0026	0.0081
5	50	50	239.05	0.0021	0.0117
5	100	10	267.15	0.0025	0.0026
5	100	30	258.88	0.0026	0.0105
5	100	50	239.05	0.0025	0.0170
10	50	10	239.05	0.0017	0.0050
10	50	30	239.05	0.0025	0.0135
10	50	50	327.15	0.0029	0.0221
10	100	10	239.05	0.0021	0.0050
10	100	30	230.78	0.0019	0.0201
10	100	50	239.05	0.0021	0.0325
20	50	10	290.78	0.0023	0.0085
20	50	30	258.88	0.0022	0.0265
20	50	50	290.78	0.0027	0.0465
20	100	10	230.78	0.0023	0.0120
20	100	30	230.78	0.0019	0.0345
20	100	50	299.05	0.0030	0.0616

Table 8 illustrates the simulation results of the ACO method used for routing in a 100-node topology. Mean and standard deviation (Std) of calculated path lengths from the sensor to the sink node are provided for each selected sensor node. The stability and effectiveness of ACO in identifying the best routes are evaluated through the simulation using multiple cycles and iterations. The average path length over all runs is represented by the mean. The standard deviation, or Std (m), indicates the variability in path selection. For all cycles, convergence to a single optimal path is demonstrated by zero Std values. Table 9 illustrates how the ACO parameters (α, β, and ρ) affect the mean and standard deviation of the path length for selected sensors.

Table 8. Performance evaluation of ACO-based routing in WSN: Mean and standard deviation of path lengths.

Sensor	Mean	Std	Sensor	Mean	Std
0	771.45	69.65	98	694.5	144
75	742.5	60	10	562.5	0
99	714.5	144	30	157.5	0
22	322.5	0	77	674.5	144
65	77.5	0	2	322.5	0
80	182.5	0	97	322.5	0
14	122.5	0	49	257.5	0
34	77.5	0	98	694.5	144

Table 9. Effect of parameters α, β, and ρ on mean and standard deviation measurements.

α	β	ρ	Sensor	Mean	Std	α	β	ρ	Sensor	Mean	Std
0.5	2	0.1	0	292.57	23.55	1	5	0.3	75	371.9	0
0.5	2	0.3	99	684.31	178.17	1	10	0.1	99	757.7	14.4
0.5	5	0.5	75	444.57	14.4	1	10	0.5	0	1056.93	0
0.5	10	0.1	75	635.84	199.6	2	2	0.3	75	1809.75	0
0.5	5	0.1	99	382.71	172.64	2	5	0.5	0	790.62	0
1	2	0.5	0	1585.1	0	2	10	0.1	0	578.5	0
1	5	0.3	75	371.9	0						

In order to extend this work, one can use other optimization techniques other than ant colony optimization for the sake of comparison and to show the effectiveness of suggested optimizer. One may use butterfly optimization, grey-wolf optimization, particle swarm optimization and bee optimization as recent and effective optimization techniques [37-41]. The artificial intelligence approaches can be utilized to improve this study for future works [42-48]. One can implement the proposed method in real-time environment by introducing embedded system design tools to develop the algorithm in real applications. LabVIEW software supported by NI equipment, or FPGA development kits can be used for this purpose [49- 55].

5. Conclusions

This study compares two popular algorithms: Dijkstra's algorithm and ACO, to determine their efficiency in route optimization for energy saving within WSNs. Dijkstra's algorithm performs consistently well in optimizing energy consumption within WSNs. Its deterministic nature enables precise route calculation, minimizing energy dissipation throughout the network. Dijkstra's algorithm is computationally efficient, providing an advantage in scenarios where consistency and low complexity are essential. Ant Colony Optimization, on the other hand, exhibits promising results by frequently achieving near-optimal outcomes and occasionally reaching optimal solutions.

However, its stochastic nature introduces variability in performance and computational complexity, making it challenging in real-time WSN applications where reliability and predictability are crucial. Simulation results indicate that Dijkstra's algorithm consistently outperforms ACO in energy consumption across various network sizes, particularly in smaller networks. For instance, in the 20-node topology, Dijkstra achieved an energy cost of 0.00226, while ACO ranged from

0.0019 to 0.0031. Although ACO sometimes matched Dijkstra's energy levels, it often came with increased complexity and inconsistency. In larger networks, Dijkstra maintained an energy cost around 0.0023-0.0025, whereas ACO exhibited greater variability, occasionally exceeding 0.0030. Dijkstra's execution times were rapid, typically under 0.001 seconds, which is crucial for real-time IoT applications. Future research could improve ACO through adaptive parameter tuning, energy-efficient pheromone updating, and hybrid approaches with Dijkstra's algorithm.

References

1. Alattas, R.J.; Patel, S.; and Sobh, T.M. (2019). Evolutionary modular robotics: Survey and analysis. *Journal of Intelligent and Robotic Systems*, 95(3), 815-828.
2. Villamil, S.; Hernández, C.; and Tarazona, G. (2020). An overview of internet of things. *Telkomnika* (*Telecommunication Computing Electronics and Control*), 18(5), 2320-2327.
3. Ray, P.P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3), 291-319.
4. Al-Ani, K.W.; Abdalkafor, A.S.; and Nassar, A.M. (2020). An overview of wireless sensor network and its applications. *Indonesian Journal of Electrical Engineering and Computer Science*, 17(3), 1480-1486.
5. Javaid, M.; Haleem, A.; Rab, S.; Singh, R.P.; and Suman, R. (2021). Sensors for daily life: A review. *Sensors International*, 2, 100121.
6. Nourildean, S.W.; Hassib, M.D.; and Mohammed, Y.A. (2022). Internet of things based wireless sensor network: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, 27 (1), 246-261.
7. Alhasan, R.A.; and Hamza, E.K. (2023). A novel CNN model with dimensionality reduction for WSN intrusion detection. *Revue d'Intelligence Artificielle*, 37(5), 1121-1131.
8. Pantazis, N.A.; Nikolidakis, S.A.; and Vergados, D.D. (2012). Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 15(2), 551-591.
9. Zhang, Y.; Sun, L.; Song, H.; and Cao, X. (2014). Ubiquitous WSN for healthcare: Recent advances and future prospects. *IEEE Internet of Things Journal*, 1(4), 311-318.
10. Haseeb, K.; Ud Din, I.; Almogren, A.; and Islam, N. (2020). An energy efficient and secure IoT-based WSN framework: An application to smart agriculture. *Sensors*, 20(7), 2081.
11. Hamza, E.K.; Salman, K.D.; and Jaafar, S.N. (2023). *Wireless sensor network for robot navigation*. In Azar, A.T., Kasim Ibraheem, I., Jaleel Humaidi, A. (Eds.), *Mobile robot: Motion control and path planning*. *Studies in computational intelligence*, Springer, Cham, 643-670.
12. Hamza, E.K.; and Alhayani, H.H. (2018). Energy consumption analyzing in single hop transmission and multi-hop transmission for using wireless sensor networks. *Al-Khwarizmi Engineering Journal*, 14 (1), 156-163.
13. Ramadhan, N.M.; Majeed, A.M.; and Raafat, S.M. (2022). A survey of power consumption minimization in WSN using feedback control strategies. *Iraqi*

Journal of Computers, Communications, Control and Systems Engineering, 22 (4), 37-47.

14. Yan, X.; Huang, C.; Gan, J.; and Wu, X. (2022). Game theory-based energy-efficient clustering algorithm for wireless sensor networks. *Sensors*, 22(2), 478.
15. Kaur, S.; and Mahajan, R. (2018). Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egyptian Informatics Journal*, 19(3), 145-150.
16. Abderrahim, M.; Hakim, H.; Boujemaa, H.; and Touati, F. (2019). Energy-efficient transmission technique based on dijkstra algorithm for decreasing energy consumption in WSNs. *Proceedings of the 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Sousse, Tunisia, 599-604.
17. Zhang, X.; Zhang, X.; and Han, L. (2019). An energy efficient Internet of Things network using restart artificial bee colony and wireless power transfer. *IEEE Access*, *7*, 12686-12695.
18. Jain, S.; and Agrawal, N. (2020). Development of energy efficient modified LEACH protocol for IoT applications. *Proceedings of the 12th International Conference on Computational Intelligence and Communication Networks* (*CICN*), Bhimtal, India*,* 160-164.
19. Bhola, J.; Soni, S.; and Cheema, G.K. (2020). Genetic algorithm based optimized leach protocol for energy efficient wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 11(3), 1281-1288.
20. Prajapat, R.; Yadav, R.N.; and Misra, R. (2021). Energy-efficient k-hop clustering in cognitive radio sensor network for internet of things. *IEEE Internet of Things Journal*, 8(17), 13593-13607.
21. Abdulzahra, A.A.; Khudor, A.Q.; and Alshawi, I.S. (2023). Energy-efficient routing protocol in wireless sensor networks based on bacterial foraging optimization. *Indonesian Journal of Electrical Engineering and Computer Science*, 29(2), 911-920.
22. Hamza, M.A. et al. (2023). Energy-efficient routing using novel optimization with tabu techniques for wireless sensor network. *Computer Systems Science and Engineering*, 45(2), 1711-1726.
23. Gunigari, H.; and Chitra, S. (2023). Energy efficient networks using ant colony optimization with game theory clustering. *Intelligent Automation and Soft Computing*, 35(3).
24. Khudhur, D.D.; and Croock, M.S. (2021). Application of self-managing system in greenhouse with wireless sensor network. *Iraqi Journal of Computer, Communication, Control and System Engineering*, 21(1), 53-61.
25. Abdulaziz, W.B.; and Croock, M.S. (2022). Design of smart irrigation system for vegetable farms based on efficient wireless sensor network. *Iraqi Journal of Computer, Communication, Control and System Engineering* (*IJCCCE*), 22(1), 72-85.
26. Heinzelman, W.R.; Chandrakasan, A.; and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, 1-10.

27. Hamza, E.K.; Mohammed, L.A.; and Hasan A.M. (2024). Dynamic parameter adjustment in ant colony optimization for energy efficiency in wireless sensor network. *Journal of Engineering Science and Technology* (*JESTEC*), 19(6), 2225-2249.
28. Razzaq, M.; Kwon, G.R.; and Shin, S. (2018). Energy efficient Dijkstra-based weighted sum minimization routing protocol for WSN. *Proceedings of the 3rd International Conference on Fog and Mobile Edge Computing* (*FMEC*), Barcelona, Spain, 246-251.
29. Deepa, G.; Kumar, P.; Manimaran, A.; Rajakumar, K.; and Krishnamoorthy, V. (2018). Dijkstra algorithm application: shortest distance between buildings. *International Journal of Engineering and Technology*, 7(4.10), 974-976.
30. Verma, D.; Messon, D.; Rastogi, M.; and Singh, A. (2021). Comparative study of various approaches of Dijkstra algorithm. *Proceedings of the International Conference on Computing, Communication, and Intelligent Systems* (*ICCCIS*), Greater Noida, India, 328-336.
31. Ojekudo, N.A.; and Akpan, N.P. (2017). An application of Dijkstra's algorithm to shortest route problem. *IOSR Journal of Mathematics* (*IOSR-JM*), 13(3), 14.
32. Luo, M.; Hou, X.; and Yang, J. (2020). Surface optimal path planning using an extended Dijkstra algorithm. *IEEE Access*, 8, 147827-147838.
33. Manev, N.; Temelkovski, B.; Serafimova, N.; and Achkoski, J. (2020). Novel approach for finding shortest route using Dijkstra's algorithm and fuzzy logic in a wireless sensor network integrated in a forest fire detection system. *Environmental Engineering and Management Journal*, 19(6), 1007-1016.
34. Dorigo, M.; Maniezzo, V.; and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41.
35. Mahmood, N.S.; Humaidi, A.J.; and Al-Azzawi, R.S. (2023). Nonlinear PD state feedback control for electronic throttle valve based on ant colony optimization. *Proceedings of the 11th Conference on Systems Process and Control* (*ICSPC*), Malacca, Malaysia, 38-43.
36. Mohammed, H.; Ibrahim, M.; Raoof, A., Jaleel, A.; and Al-Dujaili, A.Q. (2025). Modified ant colony optimization to improve energy consumption of cruiser boundary tour with internet of underwater things. *Computers* (*2073-431X*), 14(2), 74.
37. Al-Qassar, A.A.; Abdulkareem, A.I.; Humaidi, A.J.; Ibraheem, I.K.; Azar, A.T.; and Hameed, A.H. (2021). Grey-wolf optimization better enhances the dynamic performance of roll motion for tail-sitter VTOL aircraft guided and controlled by STSMC. *Journal of Engineering Science and Technology* (*JESTEC*), 16(3), 1932-1950.
38. Abdul-Kareem, A.I. et al. (2022). Rejection of wing-rock motion in delta wing aircrafts based on optimal LADRC schemes with butterfly optimization algorithm. *Journal of Engineering Science and Technology* (*JESTEC*), 17(4), 2476-2495.
39. Al-Qassar, A.A.; Al-Dujaili, A.Q.; Hasan, A.F.; Humaidi, A.J.; Ibraheem, I.K.; and Azar, A.T. (2021). Stabilization of single-axis propeller-powered system

for aircraft applications based on optimal adaptive control design. *Journal of Engineering Science and Technology* (*JESTEC*), 16(3), 1851-1869.

40. Yousif, N.Q.; Hasan, A.F.; Shallal, A.H.; Humaidi, A.J.; and Rasheed, L.T. (2023). Performance improvement of nonlinear differentiator based on optimization algorithms. *Journal of Engineering Science and Technology* (*JESTEC*), 18(3), 1696-1712.
41. Kadhim, R.A.; Kadhim, M.Q.; Al-Khazraji H.; and Humaidi A.J., (2024). Bee algorithm based control design for two-links robot arm systems. *IIUM Engineering Journal*, 25(2), 367-380.
42. Korial, A.E; Gorial, I.I.; and Humaidi, A.J. (2024). An improved ensemble-based cardiovascular disease detection system with chi-square feature selection. *Computers*, 13(6):126.
43. Abed, R.A.; Hamza, E.K.; and Humaidi, A.J. (2024). A modified CNN-IDS model for enhancing the efficacy of intrusion detection system. *Measurement*: *Sensors*, 35, 101299.
44. Mansoor, M.I.; Tuama, H.M.; and Humaidi, A.J. (2025). Application of correlation-based recurrent neural network in porosity prediction for petroleum exploration. *Engineering Research Express*, 7(1), 015241.
45. Hady, H.N.; Hadi, R.H.; Hassoon, O.H.; Hasan, A.M.; and Humaidi, A.J. (2025). Predicting process quality in multi-stage manufacturing using AE-BilA: An autoencoder-BiLSTM with attention mechanism. *Engineering Research Expres*s, 7(1), 015424.
46. Samaan, S.S; Korial, A.E.; Sarra, R.R.; and Humaidi, A.J. (2025). Multilingual web traffic forecasting for network management using artificial intelligence techniques. *Results in Engineering*, 105262.
47. Ahmad, I.A.; Hasan, A.M.; and Humaidi, A.J. (2025). Development of a memory-efficient and computationally cost-effective CNN for smart waste classification. *Journal of Engineering Research*.
48. Nasser, A.R.; Azar, A.T.; Humaidi, A.J.; Al-Mhdawi, A.K.; and Ibraheem. I.K. (2021). Intelligent fault detection and identification approach for analog electronic circuits based on fuzzy logic classifier. *Electronics*, 10(23), 2888.
49. Humaidi, A.J.; Kadhim, T.M.; Hasan, S.; Ibraheem, K.I.; and Azar, A.T. (2020). A generic Izhikevich-modelled FPGA-realized architecture: A case study of printed English letter recognition. *Proceedings of the 24th International Conference on System Theory*, *Control and Computing,* (*ICSTCC*), Sinaia, Romania, 825-830.
50. Dalloo, A.M.; and Humaidi, A.J. (2024). Optimizing machine learning models with data-level approximate computing: The role of diverse sampling, precision scaling, quantization and feature selection strategies. *Results in Engineering*, 24, 103451.
51. Dalloo, A.M.; Humaidi, A.J.; Mhdawi, A.K.A.; and Al-Raweshidy, H. (2024). Low-power and low-latency hardware implementation of approximate hyperbolic and exponential functions for embedded system applications. *IEEE Acces*s, 12, 24151-24163.
52. Dalloo, A.M.; Humaidi, A. J; Al Mhdawi, A.K., and Al-Raweshidy, H. (2024). Approximate computing: concepts, architectures, challenges, applications, and future directions. *IEEE Access*, 12, 146022-146088.

53. Jirjees, S.W.; Alkhalid, F.F.; Hasan, A.M.; and Humaidi, A.J. (2025). A secure password based authentication with variable key lengths based on the image embedded method. *Mesopotamian Journal of Cybersecurity*, 5(2), 491-500.
54. Humaidi, A.J.; and Kadhim, T.M. (2018). Spiking versus traditional neural networks for character recognition on FPGA platform. *Journal of Telecommunication*, *Electronic and Computer Engineering* (*JTEC*), 10(3), 109-115.
55. Muhammed, M.L.; Flaieh, E.H.; and Humaidi, A.J. (2022). Embedded system design of path planning for planar manipulator based on chaos A* algorithm with known-obstacle environment. *Journal of Engineering Science and Technology* (*JESTEC*), 17(6), 4047-4064.